Exploring Clustered Financial Data

Keith Johansen

Spring 2009

Since I was not sure about Carmen size limits you can find all the code and accompanying data at my Google Code repository: http://code.google.com/p/cse694-stock-cluster/source/browse/#svn/trunk. A makefile is provided. To run the program pass the final data through a pipe to the executable, for instance cat finaldata.csv | ./stock-vis

1 Introduction

1.1 Problem Definition

In financial analysis it is common to group stocks or assets and to select assets from each of the formed groups in order to diversify your holdings, assuming that different groups react to the market in different ways. That way no matter what the movement of the market:up, down, neutral, you hope to make money or at least protect yourself from massive losses. This is most common in the diversification of a mutual fund or a personal retirement account. This grouping can be done in many ways. Common ways are to group by industry, size, or growth perspective. The argument is that these diversifications are subjective and do not necessarily model how stocks react to market conditions. Thus a more principled machine learning clustering technique is proposed. I previously studied this more in depth in CSE 730 with Dr. Fosler

1.2 Motivation for Visualization

When experimenting with clustering, there are many problems which are hard to diagnose from numbers and algorithmic output alone. In previous experimentation, I had problems with degenerate clusters, instability of clusters, and determining the number of clusters to use. In CSE 730 my algorithm only had a single output: total reutrn of the portfolio over the testing period. This number could be deceptive and gave me no ability to conclude that my results would hold long term. There could have easily been some spurrious and unique relationship that was found by the algorithm and thus in true out of sample applications this could fail. Thus a way to judge the quality of the clustering was needed and visualization can provide an intuitive idea of how the clustering is performing.

2 Data

2.1 Transformations

I experimented with several transformation sin CSE 730. The two employed for this project are: the log of the weekly time series, and the weekly percent change. Each clustering was performed with 24 lags of the features, thus 24 dimensional data. [1] Thus a dimensionality reduction is required to plot the data. I used principal component analysis.

3 Implementation

3.1 Preprocessing

The preprocessing of extracting and cleaning the data is done in C#. The principal component analysis and clustering are done in Matlab/Octave. The stability calculations are done in Python to take advantage of the dynamic typing and well developed set operators.

3.2 Display

I initially started this application in Processing, but I decided to switch to C++ and OpenGL since my data files were relatively large and I never really got comfortable with the I/O of Processing.

[1] I shrank this from my presentation to allow for more time periods

3.2.1 Main Window

In the main window the stocks are plotted based on two principal components of the high dimensional input vector for the input data. Each cluster is assigned a color. The color is arbitrary and has no meaning other than as a label. Keyboard strokes can be used to zoom and scroll.

3.2.2 Stability Bar Chart Windows

The stability bar charts are an idea "inspired" by the interpretative dance group presentation. There is one window, and two bars. The red bar is the global stability that is calculated from the begining to the current time iteration. The second, green bar, is a user controlled interval. The user can increase and decrease the number of lags included by using 'c' and 'v'.

3.2.3 History Window

The history window is also an idea "inspired" by the interpretative dance group presentation. I show plots for all time steps up to the currently selected time step. The previous time steps are faded based on how far in the past they are. In this plot I no longer have outlines and solids, a faded outline was very hard to see. In the history plot I argue that the position is more important for inference than the positive or negative indication. The history window suffers from the fact that K-means clustering does not consistently label clusters. Even if a cluster stays exaclty the same during two time periods it may be arbitrarily relabeled since the cluster numebrs are meaningless labels. I would need to develop a more meaningful way to assign colors, possibly based on PCA location, to overcome this.

3.2.4 Keyboard Controls

The keyboard controls are not well though out and are thus not very intuitive.

- F2: advances a time step
- F1: decreases a time step
- Home: loops through cluster configurations
- X: zoom in
- Z: zoom out
- c: increase the number of lags in user controlled stability plot
- v: decrease the number of lags in user controlled stability plot
- Arrow keys: scroll as would be expected
- ESC: closes the program

3.3 Interesting Observations

In my CSE 730 project I identified the percent change transformation as more profitable, from examining it in the visualization I can tell that the clusters were very haphazard and not well defined, therefore he transoformation may not be more profitable long term.

4 Future Work

This is something that I think, if refined, could have useful purposes in my forthcoming career.

- Make the stability bar charts into boxplots, this would give more inferencing ablity
- Integrate the preprocessing into the display more smoothly
- More professional OpenGL features such as window resizing handling
- Allow selection of a stock, and watch its progression through time
- Apply other similarity measures
- Experiment with other dimensionality reduction techniques
- Explore other data transformations
- Make color assignment less arbitrary at each time step